

FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

**Per Aspera ad Astra:
On the Way to Parallel Processing**

F. Hoßfeld, W.E. Nagel

KFA-ZAM-IB-9507

April 1995
(Stand 13.04.95)

Erscheint in den Proceedings des Seminars Supercomputing '95 in Mannheim, 22.-24. Juni 1995,
K.G. Saur Verlag, München

Per Aspera ad Astra: On the Way to Parallel Processing

F. Hossfeld and W. E. Nagel

Central Institute for Applied Mathematics
Research Centre Jülich (KFA)
D-52425 Jülich, Germany

Abstract. Computational Science and Engineering is being established as a third category of scientific methodology; this innovative discipline supports and supplements the traditional categories: theory and experiment, in order to solve the problems arising from complex systems challenging science and technology. While the successes of the past two decades in scientific computing have been achieved essentially by the technical breakthrough of the vector-supercomputers, today the discussion about the future of supercomputing is focussed on massively parallel computers. The discrepancy, however, between peak performance and sustained performance achievable with algorithmic kernels, software packages, and real applications is still disappointingly high. An important issue are programming models. While Message Passing on parallel computers with distributed memory is the only efficient programming paradigm available today, from a user's point of view it is hard to imagine that this programming model, rather than Shared Virtual Memory, will be capable to serve as the central basis in order to bring computing on massively parallel systems from a sheer computer science trend to the technological breakthrough needed to deal with the large applications of the future; this is especially true for commercial applications where explicit programming the data communication via Message Passing may turn out to be a huge software-technological barrier which nobody might be willing to surmount.

KFA Jülich is one of the largest big-science research centres in Europe; its scientific and engineering activities are ranging from fundamental research to applied science and technology. KFA's Central Institute for Applied Mathematics (ZAM) is running the large-scale computing facilities and network systems at KFA and is providing communication services, general-purpose and supercomputer capacity also to the HLRZ ("Höchstleistungsrechenzentrum") established in 1987 in order to further enhance and promote computational science in Germany. Thus, at KFA - and in particular enforced by ZAM - supercomputing has received high priority since more than ten years. What particle accelerators mean to experimental physics, supercomputers mean to Computational Science and Engineering: Supercomputers are the accelerators of theory!

1 The Tripod of Scientific Methodology

During the past decades, modelling and computer simulation, more comprehensively identified as Computational Science and Engineering, has grown and established itself as the third category of scientific methodology, more comprehensively identified as Computational Science and Engineering. This innovative discipline fundamentally supports and supplements theory and experiment, as the two traditional categories of scientific investigation, in a qualitative and quantitative manner while integrating these into the *methodological tripod* of science and engineering (Fig. 1). Its main instrument is the supercomputer; its primary technique is computer simulation¹. The various strategic position papers²⁻⁴ and government technology programs in the U.S., in Europe, and Japan claim that the timely provision of supercomputers to science and engineering and the ambitious development of innovative supercomputing hardware and software architectures as well as new algorithms and effective programming tools are an urgent research-strategic response to grand challenges arising from huge scientific and technological barriers⁵.

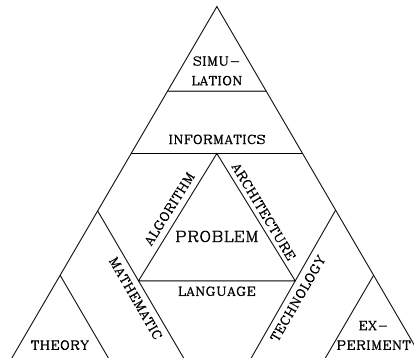


Figure 1: Problem solving and the tripod of scientific methodology

Thus, this tripod has proved to provide scientific research and technology with the stable methodological basis and the instrumental laboratory to effectively approach the solutions of the complex problems which are crucial to the future of science, technology, and society. It is essential to recognize that the scientific knowledge and the technical skills, which are available in the field of supercomputers and their applications and which will be further gained from scientific and technical engineering projects within universities and research institutions, will be a crucial factor for the industry in order to meet the requirements of international economic competition especially in the area of high-tech products. Despite the remarkable investments in research centers and universities in building up supercomputing power and skills and also some sporadic efforts in the industry

concerning supercomputing in Europe, it took until the 90s that the U.S. and European as well as national governments started non-military strategic support programs like HPCC, HPCN, and HPSC⁶⁻⁸. Their goals are also to enhance supercomputing as an innovative technology in science and engineering by stimulating the technology transfer from universities and research institutions into industry and by increasing the fraction of the technical community which gets the opportunity to develop the skills required to efficiently access the high-performance computing resources.

2 The Research Fields of KFA Jülich

The Research Centre Jülich (KFA) is one of the largest big-science centres in Europe carried by the German Federal Government and the local State Government. Today, KFA takes the function and character of a national research laboratory with highly interdisciplinary research and manifold national and international interactions and cooperations with universities, research institutes, and industry. Its research and engineering activities are focussing on five research areas: properties of matter, information technology, energy, environment, and life sciences. Computational Science and Engineering has received high recognition and priority at KFA since many years. In addition, in 1987 the Supercomputer Centre for Science and Research in Germany (in German named Höchstleistungsrechenzentrum: HLRZ) was established in its major part at KFA in an initiative similar to NSF's in the U.S. HLRZ is carried by KFA, GMD, and DESY; its primary mission is to provide supercomputer capacity for large research projects to the German science community.

3 ZAM's Mission and Super-Computational Background

The Central Institute for Applied Mathematics (ZAM) at KFA Jülich is responsible for the planning, installation, management, and operation of the central computer systems and of the KFA-wide computer networks. Its mission as a central institute and the needs for scientific services at KFA define ZAM's research and development projects in the fields of mathematics, computing, and computer science. ZAM also runs the supercomputer systems as provided to HLRZ by KFA; presently, about 150 approved projects in computational science are granted via HLRZ on the supercomputers at KFA spreading these invaluable resources over universities and research institutions throughout Germany.

Since the 60s, ZAM has run one of the most powerful scientific computing centres in Europe. At present, two vector-supercomputers CRAY Y-MP/M94 and Y-MP8/864 and a massively parallel system Intel Paragon XP/S10, a central unix

server IBM SP2 as well as an IBM mainframe ES/9000-620 together with a host of smaller systems for special purposes like visualization and communications are available to the users. Supporting the various information processing tasks in a scientific-technical environment requires more than providing access to powerful computer systems, large storage capacity and adequate software functions. To meet these needs, ZAM provides and operates various data communication networks with the following functions: (1) interactive access to information services and remote computers using workstations, PCs, or terminals at the work-place; (2) computer-computer communication inside KFA and with external data processing systems. The KFA-wide network KFAnet, a fast Local Area Network based on Ethernet and FDDI (optical fibre backbone), is open to all institutes and organizational units at KFA. Another important service is the access to the public networks of the German Telekom and to the scientific wide-area networks like Internet and the Science Network (WiN).

Facing the utmost importance of data communication in the scientific-technical environment at KFA, the services must be capable to guarantee a continuously and efficiently running network; there must also be the skill to master the underlying communication techniques, and, if necessary, to develop components to serve requirements which commercial systems do not meet. Such developments and the exploration of new communication methods and techniques are subjects of corresponding research and development projects of ZAM.

A major permanent task is to provide suitable programming languages and tools, to investigate programming methodologies, to maintain basic software, especially in the application fields of graphics, text processing, and databases, and to support users in solving complex programming problems. The programming language Fortran is still predominant in scientific and technical fields. Activities concentrate on adequate compilers and debuggers, on teaching the language and programming techniques, and on assessing future developments of this language in particular with respect to supercomputing. In the area of programming techniques and tools, the methodology of programming aims at the optimization of programs. ZAM provides and, to a great deal, also develops tools for these requirements. Supercomputer simulations need efficient visualization techniques; therefore, ZAM develops software oriented along the user requirements utilizing high-performance visualization systems. Time-dependent processes can be visualized by means of techniques established in a Video Lab.

ZAM works closely together with many other KFA institutes and HLRZ projects to select or develop appropriate solution methods and mathematical software, especially in the area of mathematical modelling, algorithm and program development and optimization, evaluation of hardware and software as well as

support of large-scale applications. Intensive collaborations, partly by contract, exist with manufacturers. ZAM is also involved in international and national research cooperations with academia and industry funded by the European Union and the German Ministry for Research and Technology.

The ZAM Information Centre is the active interface to the users. ZAM also offers classes, training courses, and seminars on many fields of information processing and computing.

4 The Structural Pyramid of Modern Computing Infrastructure

For the first time in computing history, we will be able to build a balanced pyramid of computing power in scientific and technical computation in which each element of the pyramid supports the others ⁹. At the apex of the pyramid will be the highest level of compute power which can be realized by the computer architects and the industry with respect to efficient hardware and software targeting at the teraflops systems requested by the Grand Challenges. This implies that, as a lower level of the pyramid and in order to develop the skills and the applications of future innovative computer architectures, universities and research institutions as well as industrial research divisions should be provided with mid-sized supercomputer systems. This level is required for the demanding science and engineering problems that do not need the very maximum of computing capacity, and for the computer science and computational mathematics community in order to take care of the architectural, operating systems, tools, and algorithmic issues. A third and, according to the structure of the pyramid, much broader level of scientific computing has to be supported by powerful workstations as the effective workbenches of scientists and engineers, in addition to the tremendous functionality of personal computers.

These facilities have to be networked campus-wide or corporate-wide with easy access to external communication services like Internet, which leads to the very basement of the pyramid - the network. High-speed communication with broadband functionality is promoted in the U.S. on a large scale for scientific as well as commercial applications and also in some european countries strong efforts are made to provide the scientific community with broadband communication services, e.g. in Great Britain with SuperJanet. Other european countries are either still quite far from having access to broadband communications or just start to establish test-beds with innovative technologies like ATM. In Germany, due to the high license costs involved, so far many universities could not manage to get interconnected to the German Science Network (WiN) with a maximum transmission rate of 2 mbps available since years. The backlash in high-speed communications is a severe barrier to a nation-wide infrastructure capable to provide supercomputer power and functionality to the scientific community on a large scale with transfer opportunities into the industry ¹⁰.

5 Vector-Parallel Supercomputing

Partial differential equations have been dominating in the advancement of high-speed computers and in the exploitation of their potential: "numerical windtunnel". The solution methodology for such equations leads via discretization into linear algebra and its numerical concepts and algorithms. The response of computer architecture to these early challenges of PDEs have been the vectorcomputers optimizing vector-pipeline processing and creating the effective instruments of vectorization ¹¹. Already in 1982, however, Cray Research made the significant step into multiprocessor vector-architectures and, hence, into parallel processing. On Cray multiprocessor systems, there are three different concepts which support parallelism on the programming language level: macrotasking, microtasking, and autotasking. The autotasking concept provides a user-friendly interface which can be used efficiently for fine-grain parallelism and removes some of the limitations which have restricted the usage of multitasking in the past ¹². Based on highly optimized library routines, more than 60% or even 80% of the peak performance can be obtained for large scientific applications ¹³⁻¹⁴.

While the efficiency of multitasking is proven for parallel programs running in dedicated environments since many years, the interest was focussing more and more on information about the effects introduced by parallel programs running in multiprogramming environments. With the development of the benchmark control system PARbench ¹⁵, which enables measurements of effects introduced by parallel programs running in a multiprogramming mode, we were able to evaluate and to compare different multitasking implementations, different operating systems, and different computer hardware. Moreover, we have studied scheduling algorithms, and it has been shown that efficient scheduling algorithms on the operating system level have to cooperate with lower-level work distribution algorithms on the application level which have detailed information about program structure and parallelism requirements ¹⁶.

The exploration of the computing potential of the pipelining principle including programming and compiler techniques, tools, operating system functionality, and shared-memory organization and optimization resulted in an efficient arsenal of knowledge and experience about vectorcomputing. Certainly, vectorcomputers will further develop in functionality and performance towards the 100 gigaflops target by exploiting the architectural and technological potential and expanding the "weak" parallelism well beyond the available number of processors. Even today, in the end, the sustained performance of these systems, e.g. CRAY C-90, NEC SX-3, or Fujitsu VPP500, turns out to be still ahead of the sustained performance of massively parallel systems for a vast majority of essential applications. Therefore, despite the relative progress of massively parallel computers, the very workhorses of Computational Science and Engineering are still vector-supercomputers ¹⁷.

6 The Push towards Massive Parallelism: The Software Issue

Workstations and - despite the crucial communication issue - workstation clusters, on the other hand, provide the excellent capacity to free the higher-class supercomputers from the increasing number of "small" supercomputer applications by off-loading, thus reserving them for the really large applications of the Grand Challenge category which can justify the high expenditures. However, massively parallel computers are undoubtedly considered as the - only - remedy to the needs of the demanding applications in the Grand Challenge category and maybe yet unrecognized commercial applications.

Unfortunately, in the early 90s the manufacturers of massively parallel systems promised that they would be capable to develop and deliver parallel supercomputers in 1995 which easily be able to reach the magical "3 T's" (i.e. Teraflops in execution rate, Terabyte in main memory, and Terabyte/s interconnection bandwidth), thus indicating rather a revolutionary than evolutionary step of almost three orders of magnitude beyond the then valid state-of-the-art supercomputer performance. In the meanwhile, there has not only happened a shake-out in the respective computer industry. Many investments into this massively parallel computing strategy may be definitely lost; the establishment of a new hardware and software platform will require new investments concerning finances and manpower as well as psychological recovery from the frustration caused by the unfulfilled promises of manufacturers.

6.1 The Paragon Experience

In 1991, KFA entered the field of massively parallel computing by joining an external partnership with Intel SSD and acquiring, in the end, a 140 node Paragon XP/S10. This system is a distributed-memory scalable multicomputer ¹⁸ running OSF/1 as the default operating system; the processing nodes, which are based on the Intel i860 XP RISC processor, are connected by a two-dimensional mesh with relatively high bandwidth (measured: about 90 MByte per second; specification: 200 MByte per second), and the peak performance is about 10.8 GFLOPS. As the utilization of the system at KFA is quite impressive (more than 60%), the users already have accepted massively parallel computers as a tool for scientific problem solving.

A major objective in the cooperation with Intel is the development and enhancement of functions to integrate this supercomputer into the production environment of KFA, and to establish parallel supercomputing as a standard offering for scientific computing. So far, the hardware of the Paragon has reached a certain maturity; nevertheless, based on software issues the usability, availability, and stability of the system still has to be improved: The target of one reboot per week has not yet been reached; at least for the long term, a higher reboot rate is unacceptable

for computer systems running in a production mode, and at our site a tolerable value was only achieved by a massive personal investment, in particular also in helping our users in porting applications to this machine.

A key issue in massively parallel computing is scalability. Parallelizing "dusty" decks from industry is certainly an important task to do in order to increase the acceptance of parallel computing in commercial environments. However, one cannot expect terrific performance gains in many of these programs just from porting such originally sequential, in many cases also organically grown, codes to parallel systems. There is a big discrepancy between the peak rates of massively parallel systems and the sustained performance ¹⁹. With kernels, the state of the art of massively parallel computers delivers, together with a pretty large variance in the performance data depending on the definite architecture of the system and the algorithm as well, in the average around disappointing 10% of the peak rate as sustained performance. Hence, the price/performance ratio of massively parallel computers is losing part of its attractiveness, too, if compared with vector computers.

So far, microprocessor chips have been developed with a different market goal in mind. It is extremely difficult to exploit the performance hidden in the hardware design of these processors via high-level programming languages and compiler techniques ²⁰⁻²¹; very often this leads to a loss by a factor of five to ten referred to peak performance of the node ²²⁻²³. It cannot be accepted as a reasonable software-technological approach to switch back to the very old times of assembler programming to reach reasonable performance levels. Convergence of hardware and compiler design together with the development of valuable programming tools must become the future development strategy.

6.2 Tools for Parallel Programming

On parallel computer architectures, software support is still one of the major obstacles to open the usage of such systems to a broader range of applications. Experience has shown that user-friendly tools supporting, for instance, the performance analysis and debugging process are extremely helpful and can drastically shorten the time-to-solution for a given problem. At KFA, we have developed the X Window based PARvis environment which - on a post-mortem basis - translates a given trace file generated on massively parallel systems like Intel Paragon or CRAY T3D into a variety of graphical system views which provide a reasonable basis for system understanding and program optimization ²⁴. The statistics features in combination with the timeline displays are the strength of the system; based on the extremely flexible zooming and scrolling function in the timeline displays, analysis operations are supported which can significantly improve the understanding of observed performance problems.

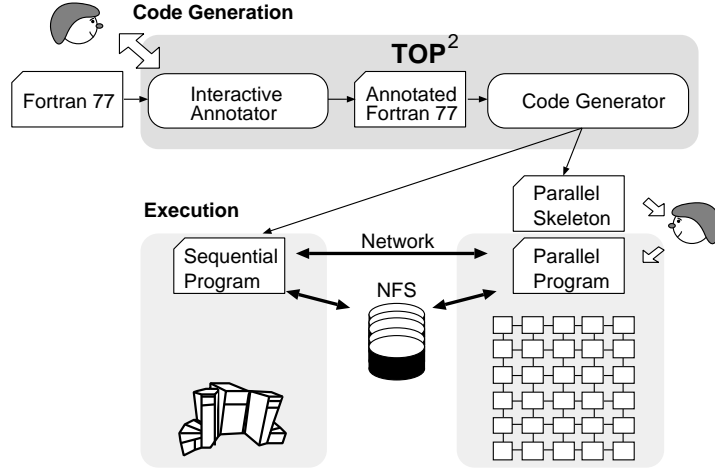


Figure 2: TOP² structure

Another worthwhile effort is the parallelization of existing sequential applications for distributed memory parallel systems, as parallelization for this type of machines is by its nature not a local operation. The tool suite TOP² developed at KFA assists users of such parallel systems in porting existing sequential Fortran applications by supporting the separation of compute-intensive kernels out of the sequential code and providing a development environment for the parallelization of these code segments²⁵. Thus, the parallelization of large applications can be broken up into several smaller tasks that may be regarded, in a sense, as local optimization steps. In this scenario, the remaining sequential and the parallel code are run simultaneously as a distributed application on both systems and automatically exchange context data between both components (Fig. 2). Main features in this process are the provision of cross-domain message passing for the automatic distribution of program data from the sequential machine (Sun, CRAY Y-MP) to the distributed memory system (Intel Paragon, CRAY T3D) and the ability of on-line debugging of the parallelized code. The data distribution features of TOP² are a subset of those defined in HPF Fortran and thus especially support algorithms on regular data structures exploiting data parallelism in the context of SPMD programming.

Nevertheless, even more powerful and effective software tools have to be developed within the next few years which will be able to increase programmers productivity by reasonable factors. More concrete, effective debugging tools, performance visualization environments, and multi-user operating systems have to be developed to extend the efficient usage of massively parallel systems.

6.3 Message Passing versus Shared Virtual Memory

Another important issue is programming models. While Message Passing is widely and effectively used on distributed memory systems as the only efficiently implemented programming paradigm at present, one can hardly imagine that this programming model will carry all future efforts to introduce massively parallel computing as the unique future technology. Despite the failure of the first commercially available massively parallel computer system which supported the programming paradigm of the Shared Virtual Memory (SVM), the efforts on this programming model should continue, because from a user's point of view, but also, maybe, from the language point of view, this SVM paradigm ²⁶⁻²⁷ seems to carry enough potential to overcome fundamental deficiencies which can be experienced from the Message-Passing paradigm.

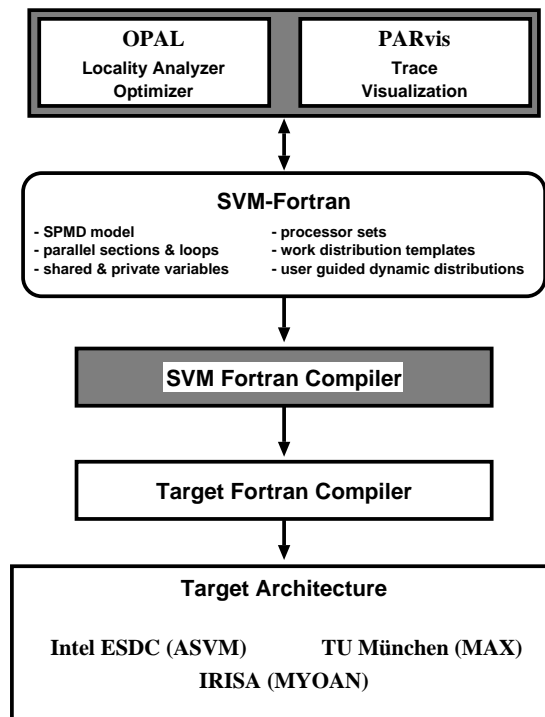


Figure 3: The SVM-Fortran environment

At KFA, we have developed SVM-Fortran as a language extension of Fortran 77 for shared-memory parallel programming on distributed memory systems. It provides special language features for optimization of data locality and load balancing. SVM-

Fortran is designed for shared virtual memory systems as well as for highly parallel computers with a hardware-based global address space. Currently, the language is translated by a source-to-source compiler into Fortran 77 with runtime library calls. Shared data structures are mapped to System V shared segments. This shared-memory interface is supported on the Intel Paragon, our current target architecture, by the Advanced Shared Virtual Memory system (ASVM) developed within a research project by the Intel European Supercomputer Development Center (ESDC). Parallelization of real-world scientific applications requires the integration of the language, the SVM implementation, as well as of programming tools into a homogeneous programming environment²⁸. Currently, the programming environment (Fig. 3) consists of the SVM-Fortran compiler, of OPAL, a source code based performance analysis tool, and the visualization tool PARvis.

First performance results demonstrate that the shared memory programming model can be efficiently used on future global address space multiprocessors via SVM-Fortran, since data locality can be enforced through well-designed work distribution annotations. However, much research and development work has to be completed to achieve production-oriented SVM implementations with tolerable overhead; in addition, powerful tools - to monitor the progress in the parallelization process - and strong support on the hardware level cannot be seen to be commercially available in the near future.

7 Conclusion: Heterogeneous (Meta)Computing as a Remedy

Researchers in the field of innovative computing believe that there will be no single all-encompassing architecture which will be capable to satisfy the heterogeneous spectrum of requirements with equally optimal performance. On the other hand, it is well known that a user tends to invest tremendous efforts in order to extract even that small level of sustained performance out of an innovative computer system for his specific application, although his heterogeneous requirements cannot be efficiently satisfied by the single target system he is focussing on just because it is available to him: *per aspera ad astra*!

The experiences with the different supercomputer architectures with their strengths and weaknesses, the technological obstacles for major performance steps in vector-computing, the large variance in the performance data for algorithms on different parallel machines quite naturally lead to the concept of heterogeneous computer systems. Heterogeneous computing is an attractive concept because it takes into account that the individual parallel machines, and vectorcomputers as well, spend much of their time on tasks for which they are unsuited; these effects lead to the experienced breakdowns in sustained performance and also to scalability problems. On heterogeneous systems ²⁹, the computational work of - parallel - programs can be split across different computers in order to achieve in total the

fastest possible execution, where the individual portions of the work are sent to those computer systems which have been proved to be best for the specific characteristics of the work. The goal of heterogeneous computing is the efficiency of computation and thereby the effectiveness and cost-effectiveness of both computers and programmers; simultaneously, it is paving the way to metacomputing.

References

1. Special Double Issue: Grand Challenges to Computational Science, Future Generation Computer Systems **5** (1989), No. 2&3.
2. Committee on Physical, Mathematical, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, Grand Challenges 1993: High Performance Computing and Communications, The FY 1993 U.S. Research and Development Program, Office of Science and Technology Policy, Washington, 1992.
3. Board on Mathematical Sciences of the National Research Council (USA), The David II Report: Renewing U.S. Mathematics - A Plan for the 1990s, in: Notices of the American Mathematical Society, May/June 1990, 542-546; September 1990, 813-837; October 1990, 984-1004.
4. Commission of the European Communities, Report of the EEC Working Group on High-Performance Computing (Chairman: C. Rubbia), February 1991.
5. U. Trottenberg et al., Situation und Erfordernisse des wissenschaftlichen Höchstleistungsrechnens in Deutschland - Memorandum zur Initiative High Performance Scientific Computing (HPSC), Februar 1992, published in: Informatik-Spektrum **15** (1992), H. 4, 218
6. The Congress of the United States, Congressional Budget Office, Promoting High-Performance Computing and Communications, Washington, June 1993.
7. High-Performance Computing Applications Requirements Group, High-Performance Computing and Networking, Report, European Union, April 1994. - High-Performance Networking Requirements Group, Report, European Union, April 1994.
8. Bundesministerium für Forschung und Technologie, Initiative zur Förderung des parallelen Höchstleistungsrechnens in Wissenschaft und Wirtschaft, BMFT, Bonn, Juni 1993.
9. NSF Blue Ribbon Panel on High Performance Computing, From Desktop To Teraflop: Exploiting the U.S. Lead in High Performance Computing, Report, NSF-CISE, 19 October 1993.
10. R. Bayer, Plädoyer für eine Nationale Informations-Infrastruktur, Informatik-Spektrum **17** (1994), H. 5, 302.
11. K. Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, 1993

12. F. Hossfeld, R. Knecht, and W. E. Nagel, Multitasking: Experience with Applications on a CRAY X-MP, *Parallel Computing* 12 (1989), No. 3, 259.
13. S. Knecht, E. Laermann, and W.E. Nagel, Parallelizing QCD with Dynamical Fermions on a CRAY Multiprocessor System, *Parallel Computing* 15 (1990), No. 1-3, 3.
14. J. Frank and S. Knecht, Lattice Quantum Hadrodynamics on a CRAY Y-MP, *The Journal of Supercomputing* 6 (1992), 195.
15. W. E. Nagel and M.A. Linn, Benchmarking Parallel Programs in a Multiprogramming Environment: The PAR-Bench System, In: *Advances in Parallel Computing, Vol. 8: Computer Benchmarks* (Dongarra, J., Gentzsch, W., eds.), Elsevier Science Publishers B. V. (1993), 303.
16. W. E. Nagel, Ein verteiltes Scheduler-System für Mehrprozessorrechner mit gemeinsamem Speicher: Untersuchungen zur Ablaufplanung von parallelen Programmen, *Berichte des Forschungszentrums Jülich Jül-2850* (1993).
17. J. J. Dongarra, H.-W. Meuer and E. Strohmaier (eds.), *TOP500 Report 1993; TOP500 Report 1994*, University of Mannheim, Germany, May/November 1994.
18. R. Berrendorf, H. C. Burg, U. Detert, R. Esser, M. Gerndt, and R. Knecht, Intel Paragon XP/S - Architecture, Software Environment, and Performance, *Interner Bericht KFA-ZAM-IB- 9409*, (1994).
19. D. H. Bailey et al., NAS Parallel Benchmark Results 3-94; NAS Parallel Benchmarks Update 9-94, RNR Technical Report RNR-94-006, NASA Ames Research Center, Moffett Field, California, March and September 1994.
20. D. F. Bacon et al., Compiler Transformations for High-Performance Computing, *ACM Computing Surveys* 26(1994), No. 4, 345-420.
21. H. Zima and B. Chapman, *Supercompilers for Parallel and Vector Computers*, ACM Press, 1990.
22. A. Mlynski-Wiese, Leistungsuntersuchung des iPSC/860 RISC-Knoten-Prozessors: Architekturanalyse und Programmoptimierung, *Zentralinstitut für Angewandte Mathematik, Berichte des Forschungszentrums Jülich Jül-2766*, Mai 1993.
23. A. Krumme, Performance-Analyse des DEC Alpha-Chip: Moderne Compiler-Techniken zur Programmoptimierung, *Zentralinstitut für Angewandte Mathematik, Berichte des Forschungszentrums Jülich Jül-2912*, Mai 1994.
24. W. E. Nagel and A. Arnold, Performance Visualization of Parallel Programs: The PARvis Environment, *Proceedings 1994 Intel Supercomputer Users Group (ISUG) Conference*, 24-31; Technical Report CCSF-47, Concurrent Supercomputing Consortium, Pasadena, California, May 1994.
25. U. Detert and M. Gerndt, TOP2 - Tool Suite for the Development and Testing of Parallel Applications, *Proceedings CONPAR'94* (1994), .
26. K. Li, Shared Virtual Memory on Loosely Coupled Multiprocessors, *PhD Thesis*, Yale University, September 1986.
27. R. Berrendorf, M. Gerndt, Z. Lahjomri and Th. Priol, A Comparison of Shared Virtual Memory and Message Passing Programming Techniques

- Based on a Finite Element Application, in: B. Buchberger and J. Volkert (eds.), *Parallel Processing: CONPAR 94 - VAPPVI, Lecture Notes in Computer Science Vol. 854*, Springer Verlag, 461-472.
28. M. Gerndt and R. Berrendorf, *Parallelizing Applications with SVM-Fortran*, to be published in *Proc. HPCN'95*.
 29. A. A. Khokhar et al., *Heterogeneous Computing: Challenges and Opportunities*, *IEEE Computer* **26** (1993), No. 6, 18.